Self-Repairing Neural Networks based on Neural Cellular Automata

Tommaso Paladini[1], Eva Tuba[2], and Luca Mariot[1]

[1] Semantics, Cybersecurity and Services Group, University of Twente,
Drienerlolaan 5, 7522 NB Enschede, The Netherlands
{t.paladini,l.mariot}@utwente.nl
[2] D.R. Semmes School of Science, Trinity University,
San Antonio, TX 78212, USA
etuba@ieee.org

1 Introduction

Cellular Automata (CA) are a parallel computational model in which complex global behaviors emerge from the repeated application of simple local rules. An interesting recent development is the Growing Neural Cellular Automaton (GNCA) framework introduced by Mordvintsev et al. [1], in which a neural network parameterizes the update rule of a CA operating on a 2D grid of cells. When trained appropriately, such systems are capable of regenerating a target pattern from a partially damaged or incomplete state, relying solely on local cell interactions. This self-organizing and self-repairing behavior has so far been exploited primarily in the domain of image synthesis and repair.

In this work, we propose a novel application of the GNCA framework: the self-repair of artificial neural networks subjected to adversarial weight tampering. Our work is motivated by a cybersecurity perspective, with two main applications in mind:

1. *Adversarial robustness*: as machine learning models are increasingly deployed in critical systems, the integrity of trained model parameters becomes a security concern. An attacker with access to the storage or transmission channel of a model may corrupt or selectively modify its weights, degrading performance in a targeted [2] or indiscriminate manner [3]. We investigate whether a neural CA, trained to "know" what the weight matrix of a layer is supposed to look like, could serve as a lightweight, distributed mechanism for detecting and correcting such tampering.
2. *Model Extraction*: beside adversarial attacks, another growing security concern is maintaining the confidentiality of the weights of a network, for intellectual property protection. In this case, we ask if a neural CA that knows a fraction of the weights is able to reconstruct the matrix completely (or a close approximation thereof).

2 Proposed Approach

Consider a trained feedforward neural network with one or more dense layers. Each such layer can be represented as a 2D matrix of real-valued weights[1]. Our idea is to embed this weight matrix directly into the state space of a 2D cellular automaton, treating each weight as the state of a single cell. A neural CA is then trained—after the base network has converged—to maintain and regenerate this weight pattern in the presence of localized perturbations.

More precisely, the CA operates on a grid of the same dimensions as the weight matrix. Each cell's state encodes the corresponding weight value, along with auxiliary channels used internally by the CA's update rule. At each time step, the CA applies its learned local rule, updating each cell based on the values of its spatial neighbours. When a subset of weights is corrupted by an adversary—replaced, perturbed, or zeroed out—the CA iterates until the grid converges to a close approximation of the original weight matrix. The repaired weights are then substituted back into the network, restoring predictive performance.

This approach requires no centralised knowledge of the full weight matrix at repair time: recovery relies exclusively on local interactions, which is the defining strength of the CA model. The trained CA thus acts as a distributed, implicit snapshot of the target weight pattern, encoded in the form of local update rules rather than in an explicit copy of the parameters.

Key Questions and Research Directions

As a proposal, we identify several interesting questions that this research programme must address.

Corruption model. The effectiveness of CA-based repair will depend on the nature and extent of the adversarial attack. Critical parameters include the fraction of corrupted weights, their spatial distribution across the layer (localised patch versus random scatter), and the magnitude of the perturbation. We expect the CA to generalise well to spatially localised corruptions, given that local rules are well-suited to filling contiguous missing regions, while scattered, high-magnitude perturbations may present a harder challenge.

Scalability and architecture. Extending the framework from single layer to high dimensionality multi-layer networks — as the ones that appear in contemporary, popular Transformer-based language models — raises questions about whether a single CA suffices for all layers or whether layer-specific CAs are needed. The number of auxiliary channels per cell, the neighbourhood size, and the CA architecture are all design dimensions to be explored.

[1] For simplicity, we omit the bias vector.

Theoretical connections. Interestingly, there may be connections to classical results in fault-tolerant computation [4] and error-correcting codes, where redundancy in a distributed system enables recovery from partial failures. The CA's update rule can be seen as implicitly encoding a form of distributed error correction over the weight space. Formalising this connection, for instance by relating repair capacity to information-theoretic bounds on recoverable corruption, is a promising theoretical direction.

References

1. A. Mordvintsev, E. Randazzo, E. Niklasson, and M. Levin. Growing neural cellular automata. *Distill*, 2020. doi:10.23915/distill.00023
2. , Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. . In 25th Annual *Network And Distributed System Security Symposium* (NDSS 2018). Internet Soc, 2018.
3. B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. Proceedings of the 29th *International Coference on International Conference on Machine Learning*. 2012.
4. J. von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. Shannon and J. McCarthy, editors, *Automata Studies*, pages 43–98. Princeton University Press, 1956.